xCAT Statelite Cookbook

03/09/10, 02:39:23 PM

1 Introduction

This document details a design for native xCAT NFS Root. We call the design statelite because in addition to having an NFS Root implementation, some files can be stored persistently and maintain state through reboots.

Statelite provides an efficient and flexible diskless solution because most of the OS image is NFS mounted read-only, but avconfigurable list of directories and files can be read-write. The read-write files can either be persistent across reboots, or volatile (restoring to pristine state after reboot).

Statelite offers the following advantages over xCAT's stateless (RAMdisk) implementation:

1. Some files can be made persistent over reboot. This is useful for license files or database servers where some state is needed. However, you still get the advantage of only having to manage a single image.

2. Changes to hundreds of machines can take place instantly and automatically by updating one main image. In most cases, machines do not need to reboot for these changes to take affect.

3. Ease of administration by being able to lock down an image. Many parts of the image can be read only so no modifications can transpire without updating the main image.

4. Files can be managed in a hierarchical manner. For example: Suppose you have a machine that is in one lab in Tokyo and another in London. You could set table values for those machines in the xCAT database to allow machines to sync from different places based on their attributes. This allows you to have one base image with multiple sources of file overlay.

5. Ideal for virtualization – In a virtual environment you may not want a disk image (neither stateless nor stateful) on every virtual node as it consumes memory and disk. Virtualizing with the statelite approach allows for images to be smaller, easier to manage, use less disk, less memory, and more agile.

Disadvantages

1. NFS Root requires more network traffic to run as the majority of the disk image runs over NFS. This may depend on your workload, but can be minimized.

2. NFS Root can be complex to set up as well. As more files are created in different places there are greater chances for failures. We think this flexibility is also one of the great virtues of Statelite. The image can work in nearly any environment.

1.1 Limitations

Currently, statelite has only been tested on Red Hat and Red Hat Clone environments on x86_64 platforms, and SLES11 (SuSE Linux Enterprise Server 11) on POWER systems. We have no reason to think it won't work Red Hat on POWER and SLES11 on x86_64.

Stateless does not yet support hierarchical clusters (those with service nodes).

A diskless implementation similar to statelite is available with xCAT on AIX, but it does not yet have all of the flexibility.

2 Usage

Getting started with Statelite provisioning requires that you have xCAT set up and running. Before continuing with the rest of this document, the xCAT management node must be set up, and the nodes' hardware control, resource, and type attributes must be defined as described in the other xCAT documentation.

Please report any errors to the xCAT mailing list (https://lists.sourceforge.net/lists/listinfo/xcat-user).

2.1 Lock Down main image

Any statelite image will need some files to be read/writable for individual nodes. But we encourage you to first lock down the main image. The image will be placed in /install/netboot/<os>/<arch>/<profile>/rootimg after running genimage (later in this doc).

xCAT, by default, during installation will create on the management node an /etc/exports entry that exports the /install directory read/writeable. Please change it to read-only, so it looks as follows:

```
/install *(ro,no_root_squash,sync)
```

Next, restart the NFS server:

```
service nfs restart
```

2.2 litefile

The litefile table specifies the directories and files on the statelite nodes that should be readwrite, persistent, or readonly overlay. All other files in the statelite nodes come from the readonly statelite image. See http://xcat.sourceforge.net/man5/litefile.5.html for details.

image

The first column of the table is the image name, as specified in the osimage table. This can be left empty, or specified with "ALL" to signify that this row applies to all all images.

file
The second column is the file name. This should the full path of the file. If the file is a directory, then it should be terminated with a '/'.

options
The third column contains the options of a file. xCAT supports the following:

- *Blank or* **tmpfs** *or* **ALL** – the file is readwrite and will be placed in tmpfs on the booted node. When searching for the file, the first one to be found in the litetree hierarchy will be used. When the node is rebooted, this file will be reinitialized.
- **persistent** - This means that the file is readwrite and will be persistent across reboots. If the file does not exist at first, it will be created during initialization. Every time there after the file will be left alone if it exists. (Requires the statelite table to be filled out with a spot for persistent storage).
- **ro** – file will be read only. This file will be located in the directory hierarchy specified in the litetree table. That directory will be mounted on the node the whole time the node is up (not just during the boot process), and the file will be linked to this directory. Changes made to this file on the server will be immediately seen in this file on the node.
- **con** – The contents of the pathname are concatenated onto the contents of the existing file. For this directive the searching in the litetree hierarchy does not stop when the first match is found. Con is similar to tmpfs, but all files found in the hierarchy will be concatenated to the file when found.
- **bind** - instead of using symbolic links, "mount --bind" is used to mount the file to the root image, and the permission for the file will keep the same as the original one.
- **bind,persistent** - the file will be persistent across reboots, and the file will be mounted with "--bind" option.

Sample Data for RedHat

Notice that all files are in tmpfs. This gives you an NFS root solution with no persistent storage.

```
#image,file,options,comments,disable
"ALL","/etc/adjtime",,,
"ALL","/etc/fstab",,,
"ALL","/etc/inittab",,,
"ALL","/etc/lvm/.cache",,,
"ALL","/etc/mtab",,,
"ALL","/etc/ntp.conf",,,
"ALL","/etc/ntp.conf.predhclient",,,
"ALL","/etc/resolv.conf",,,
"ALL","/etc/resolv.conf.predhclient",,,
"ALL","/etc/ssh/",,,
"ALL","/tmp/",,,
"ALL","/var/account/",,,
"ALL","/var/arpwatch",,,
"ALL","/var/cache/alchemist",,,
"ALL","/var/cache/foomatic/",,,
"ALL","/var/cache/logwatch/",,,
"ALL","/var/cache/man/",,,
"ALL","/var/cache/mod_ssl/",,,
"ALL","/var/cache/mod_proxy/",,,
"ALL","/var/cache/php-pear/",,,
"ALL","/var/cache/systemtap/",,,
"ALL","/var/empty/",,,
"ALL","/var/db/nscd/",,,
"ALL","/var/gdm/",,,
"ALL","/var/lib/dav/",,,
"ALL","/var/lib/dhcp/",,,
"ALL","/var/lib/dhclient/",,,
"ALL","/var/lib/php/",,,
"ALL","/var/lib/scsi/",,,
"ALL","/var/lib/ups/",,,
"ALL","/var/lib/random-seed",,,
"ALL","/var/lib/iscsi",,,
"ALL","/var/lib/logrotate.status",,,
"ALL","/var/lib/ntp/",,,
"ALL","/var/lib/xen/ntp",,,
"ALL","/var/lock/",,,
"ALL","/var/log/",,,
"ALL","/var/run/",,,
"ALL","/var/tmp/",,,
"ALL","/var/tux/",,,
"ALL","/opt/xcat/",,,
"ALL","/xcatpost/",,,
```

Sample Data for SLES11

```
#image,file,options,comments,disable
"ALL","/etc/inittab",,,
"ALL","/etc/lvm/.cache",,,
"ALL","/etc/mtab",,,
"ALL","/etc/ntp.conf",,,
```

```
"ALL","/etc/resolv.conf",,,
"ALL","/etc/ssh/",,,
"ALL","/etc/sysconfig/",,,
"ALL","/etc/syslog-ng/",,,
"ALL","/tmp/",,,
"ALL","/var/tmp/",,,
"ALL","/var/run/",,,
"ALL","/etc/yp.conf",,,
"ALL","/var/lib/",,,
"ALL","/var/empty/",,,
"ALL","/var/spool/",,,
"ALL","/var/lock/",,,
"ALL","/var/log/",,,
"ALL","/var/cache/",,,
"ALL","/etc/fstab",,,
"ALL","/var/adm/",,,
"ALL","/root/.viminfo",,,
"ALL","/root/.bash_history",,,
"ALL","/opt/xcat/",,,
"ALL","/xcatpost/",,,
```

Use the command

```
tabedit litefile
```

to copy/paste the above sample lines into the litefile table.

2.3 litetree

The litetree table controls where the initial content of the files in the litefile table come from, and the long term content of the "ro" files. When a node boots up in statelite mode, it will by default copy all of its tmpfs files from the /.default directory of the root image. You may decide that you want some of the files pulled from different locations that are different per node.

For example, a user may have two directories with a different */etc/motd* that should be used for nodes in two locations:

```
10.0.0.1:/syncdirs/newyork-590Madison/rhels5.4/x86_64/compute/etc/motd
10.0.0.1:/syncdirs/shanghai-11foo/rhels5.4/x86_64/compute/etc/motd
```

You can specify this in one row in the litetree table:

```
1,,10.0.0.1:/syncdirs/$nodepos.room/$nodetype.os/$nodetype.arch/
$nodetype.profile
```

When each statelite node boots, the variables in the litetree table will be substituted with the values for that node to locate the correct directory to use. Assuming that /etc/motd was specified in the litefile table, it will be searched for in all of the directories specified in the litetree table and found in this one.

You may also want to look by default into directories containing the node name first:

```
$noderes.nfsserver:/syncdirs/$node
```

The litetree prioritizes where node files are found. The first field is the priority. The second field is the image name (ALL for all images) and the final field is the mount point.

Our example is as follows:
```
1,,$noderes.nfsserver:/statelite/$node
2,,cnfs:/gpfs/dallas/
```

The two directories /statelite/$node on the node's $noderes.nfsserver and the /gpfs/dallas on the node cnfs contain root tree structures that are sparsely populated with files that we want to place in those nodes. If files are not found in the first directory, it goes to the next directory. If none of the files can be found in the litetree hierarchy, then they are searched for in /.default on the local image.

For details about the table format, see
http://xcat.sourceforge.net/man5/litetree.5.html .

2.4 Determine where state will be held

You may want some files in the image to be stored permanently, to survive reboots. This is done by entering the information into the statelite table.

The headings are as follows for this table:

```
#node,image,statemnt,comments,disable
```

An example would be:
```
"japan",,"cnfs:/gpfs/state",,,
```

All nodes in the japan node group will have their state stored in the /gpfs/state directory on the machine known as cnfs. This is true for all images, unless an image name is specified in the 2nd column.

When the node boots up, then the value of statemnt will be mounted to /.statelite/persistent. The code will then create the following subdirectory /.snapshot/persistent/<nodename>
This directory will be the root of the image for this node's persistent files.

To set the "statemnt" value, you can use variables in xCAT database, it follows the grammer in the "litetree" table, for example:
```
#node,image,statemnt,comments,disable
"hv32lpar05",,"$noderes.nfsserver:/lite/state",,
```

Note: Do not name your persistent storage directory after the node name, as this will be placed in the directory automatically. If you do, then a directory named

/state/n01 will have its state tree inside /state/n01/n01.

2.5 Policy

Ensure policies are set up correctly. When a node boots up, it queries the xCAT database to get the lite-files and the lite-tree. In order for this to work, the command must be set in the policy table to allow nodes to request it. (This should happen automatically when xCAT is installed, but you may want to verify it if you experience problems booting.)

```
chtab priority=4.7 policy.commands=litefile policy.rule=allow
chtab priority=4.8 policy.commands=litetree policy.rule=allow
```

2.6 Add Linux Distro Packages

If you haven't already done so, copy the packages from the distro media into /install. For example:

```
copycds /iso/RHEL5.2-Server-20080430.0-x86_64-DVD.iso
```

Now create a list of distro packages that should be installed into the image. You should start with the base packages in the compute template. These are required. For example:

```
mkdir -p /install/custom/netboot/sles
cd /install/custom/netboot/sles
cp /opt/xcat/share/xcat/netboot/sles/compute.sles11.pkglist
test1.pkglist
```

You can then add more packages to the test1.pkglist by editing test1.pkglist.

2.7 Add Third-Party Software

If you have additional software that you want in the image, you can add it to the distro package directory that was created in the previous step and then rerun createrepo. But if you want to keep your distro package directory pristine, with only distro files in it, you can use xCAT's otherpkgs support:

- Create a directory on your management node named "/install/post/otherpkgs/<OS>/<ARCH>/", and put your third party rpm packages into the directory:

  ```
  mkdir /install/post/otherpkgs/sles11.1/x86_64
  # put RPMs in there
  ```

- Create the directory "/install/custom/netboot/<OS>" and add the names of the RPM packages you want in the image into the .otherpkgs.pkglist file.

  ```
  vi compute.otherpkgs.pkglist
  ```

  Note: if some of these RPMs will need write permissions to files on the node, add those files or directories into the "litefile" table.

- Create the .repolist file for your own repository. The file name is: <PROFILE>.<OS>.<ARCH>.repolist in the same directory as above. The file format is:

```
Type|URL|name
```

For example, if the repository is local, you can add the repository information as:

```
plaindir|file:///install/post/otherpkgs/sles11/ppc64|
otherpkgs
```

If there's a remote repository, you can add the repository information as:

```
rpm-md|http://xcat.sourceforge.net/yum/xcat-dep/sles11/ppc64|
xcat-dep
rpm-md|http://xcat.sourceforge.net/yum/devel/core-snap|core-snap
```

2.8 Create Statelite Image

After all our tables are set up, it is time to create a base image. In this example, the operating system is RedHat 5.3, and we will name our image "test1".

Run the genimage command to create the image based on the pkglist's created above:

```
genimage
```

The command will prompt you for the necessary inputs (OS, profile name, etc)

Alternatively, you can run the OS-specific genimage command directly. For example, for Red Hat:

```
cd /opt/xcat/share/xcat/netboot/rh
./genimage –o rhels5.3 –p test1 –i eth0 –n mlx4_core,mlx4_en,igb,bnx2 –
m statelite
```

The genimage command will do several things:

1. It will create an image in `/install/netboot/<os>/<arch>/<profile>/rootimg`
2. It will create a ramdisk and kernel that can be used to boot the initial node.

One new option named "**--permission**" is added for statelite mode, you can assign user-definable permission for "/.statelite" directory. The default permission is 755, for example:

```
./genimage –o rhels5.3 –p test1 –i eth0 –n mlx4_core,mlx4_en,igb,bnx2 –
m statelite –permission 777
```

This image that you have created can be used for stateless or statelite booting.

2.9 Modify statelite image

Since the files that were now just created will be the default for all the files listed in the litefile table, you can edit the image directly by visiting the root tree in:

```
/install/netboot/<os>/<arch>/<profile>/rootimg
```

You can run chroot to make additional changes, for example, yum/zypper updates/install using the –installroot flag.

For Red Hat it is necessary is to copy some passwd files and ssh settings from xCAT into the image. Note: don't run the following commands for SLES.

```
cd /opt/xcat/share/xcat/netboot/add-on/statelite/
./add_passwd /install/netboot/<os>/<arch>/<profile>/rootimg
./add_ssh /install/netboot/<os>/<arch>/<profile>/rootimg
```

2.10 Run liteimg <os>-<arch>-<profile>

The liteimg command will modify your statelite image (the image that genimage just created) by creating a series of links. Once you are satisfied your image contains what you want it to, run "liteimg <os>-<arch>-<profile>". For example:

```
liteimg rhels5.3-x86_64-test1
```

or

```
liteimg –o rhels5.3 –a x86_64 –p test1
```

This creates 2 levels of indirection so that files can be modified while in their image state as well as during runtime. For example, a file like "<$imgroot>/etc/ntp.conf" will have the following operations done to it:

```
mkdir –p $imgroot/.default/etc
mkdir –p $imgroot/.statelite/tmpfs/etc
mv $imgroot/etc/ntp.conf $imgroot/.default/etc
cd $imgroot/.statelite/tmpfs/etc
ln –sf ../../../.default/etc/ntp.conf .
cd $imgroot/etc
ln –sf ../.statelite/tmpfs/etc/ntp.conf .
```

When finished, the original file will reside in $imgroot/.default/etc/ntp.conf. $imgroot/etc/ntp.conf will link to $imgroot/.statelite/tmpfs/etc/ntp.conf which will in turn link to $imgroot/.default/etc/ntp.conf

Note: If you make any changes to your litefile table after running liteimg then you will need to rerun liteimg again. This is because files and directories need to have the two levels of redirects created.

2.11 Set the boot state to "statelite"

You can now deploy the node:

```
nodeset <noderange> statelite=centos5.3-x86_64-test1
```

Or just run:

```
nodeset <noderange> statelite
```

This will create the necessary files in /tftpboot for the node to boot correctly.

2.12 Install the Node

Finally, reboot the node so that it boots up in statelite mode.

For x86_64 platform:

```
rpower <noderange> boot
```

Nodeset will have generated the appropriate PXE file so that the node boots from the nfsroot image. This file will look similar to the following:

```
#statelite centos5.3-x86_64-all
DEFAULT xCAT
LABEL xCAT
 KERNEL xcat/netboot/centos5.3/x86_64/all/kernel
 APPEND initrd=xcat/netboot/centos5.3/x86_64/all/initrd.gz
NFSROOT=172.10.0.1:/install/netboot/centos5.3/x86_64/all
STATEMNT=cnfs:/gpfs/state XCAT=172.10.0.1:3001 console=tty0
console=ttyS0,115200n8r
```

For POWER platform:

Run the following command to boot up the nodes into statelite mode:

```
rnetboot <noderange>
```

Nodeset will have generated the appropriate yaboot.conf-MAC-ADDRESS file so that the node boots off the nfsroot image. This file will look similar to the following:

```
#statelite sles11-ppc64-compute
timeout=5
image=xcat/netboot/sles11/ppc64/compute/kernel
        label=xcat
        initrd=xcat/netboot/sles11/ppc64/compute/initrd.gz
        append="NFSROOT=192.168.11.108:/install/netboot/sles11/ppc64/co
mpute STATEMNT= XCAT=192.168.11.108:3001 "
```

You can then use rcons or wcons to watch the node boot up.

3 New Commands

The following commands are in /opt/xcat/bin:

litefile <nodename>
Shows all the statelite files that are not to be taken from the base of the image.

litetree <nodename>
Shows the NFS mount points for a node.

Ilitefile <image name>
Shows the stateless files that will be used for a node image.

liteimg <image name>
Creates a series of symbolic links in an image that is compatible with statelite booting.

4 Statelite Directory Structure

Each statelite image will have the following directories:

```
/.statelite/tmpfs/
/.statelite/persistent/<nodename>
/.statelite/mnt  # where directory tree is mounted from.
/.default/
/etc/init.d/statelite
```

All files that are symbolic links, will link to /.statelite/tmpfs.

tmpfs files that are persistent link to /.statelite/persistent/<nodename>/
/.statelite/persistent/<nodename> is the directory where the node's individual storage will be mounted to.

/.default is where default files will be copied to from the image to tmpfs if the files are not found in the litetree hierarchy.

4.1 Options filling out tables

noderes.nfsserver can be filled out for the NFSroot server. If this is not filled out then it uses the management server.

noderes.nfsdir – can be filled out: /vol/xCAT/install. At that point it assumes that the directory structure is the same as the install directory.

5 Adding/updating software and files for the running nodes

Because most of system files for the nodes are NFS mounted on the management node with read-only option, installing or updating software and files should be done to the image. The image is located under `/install/netboot/<os>/<arch>/<profile>/rootimg` diretory.

To install or update an rpm, do the following:

1. `rpm --root /install/netboot/<os>/<arch>/<profile>/rootimg rpm_name`
2. Restart the software application on the nodes.
   `xdsh <noderange> restart_this_software`

It is recommended to follow section 2.7 (Addig third party softeware) to add the new rpm to the otherpkgs.pkglist file so that the rpm will get installed into the new image next time the image is rebuilt.
Note: The newly added rpms are not shown when running `rpm -qa` on the nodes although the rpm is installed. It will shown next time the node is reboot.

To create or update a file for the nodes, just modify the file in the image and restart any application that uses the file.

6 Debugging techniques

1. When a node boots up in staelite mode, there is a script run called statelite that is in the root directory of $imgroot/etc/init.d/statelite. This script is not run as part of the rc scripts, but as part of the pre switch root environment. Thus, all the linking is done in this script. There is a "set –x" near the top of the file. You can uncomment it and see what the script runs. You will then see lots of mkdirs and links on the console.
2. You can also set the machine to shell. Just add the word "shell" on the end of the pxeboot file of the node in the append line. This will make the init script in the initramfs pause 3 times before doing a switch_root.
3. When all the files are linked they are logged in /.statelite/statelite.log on the node. You can get into the node after it has booted and look in the /.statelite directory.